

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

#2
8-21-01
JM



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

2000年 5月18日

出願番号
Application Number:

特願2000-146867

出願人
Applicant(s):

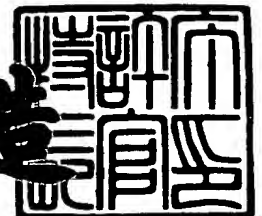
株式会社日立製作所

U. S. Appln. Filed 1-9-01
Inventor: M. Asano et al
Mathingly Stanger & Malor
Docket ASA-954

2000年12月15日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3104409

【書類名】 特許願

【整理番号】 KN1113

【提出日】 平成12年 5月18日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社
 日立製作所 システム開発研究所内

 【氏名】 浅野 正靖

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社
 日立製作所 システム開発研究所内

 【氏名】 新井 利明

【発明者】

 【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社
 日立製作所 ソフトウェア事業部内

 【氏名】 山下 洋史

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社 日立製作所

【代理人】

 【識別番号】 100078134

 【弁理士】

 【氏名又は名称】 武 顕次郎

 【電話番号】 03-3591-8550

【手数料の表示】

 【予納台帳番号】 006770

 【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 計算機システム及び計算機システムの制御方法

【特許請求の範囲】

【請求項 1】 複数のオペレーティングシステムを 1 台の計算機内で稼働させ、CPU、主記憶装置、外部入出力装置等の計算機資源を前記複数のオペレーティングシステムのそれぞれに割り当てる手段を有する計算機システムにおいて、前記計算機資源を管理する手段と、前記計算機資源の前記各オペレーティングシステムへの割り当ての変更と再構成とを行う手段と、前記各オペレーティングシステムの稼働状態に関連して前記計算機資源の変更と再構成との内容を管理する手段と、前記各オペレーティングシステムの稼働状態に基づいて前記計算機資源を変更または再構成する手段とを備えることを特徴とする計算機システム。

【請求項 2】 複数のオペレーティングシステムを稼働させ、オペレーティングシステムの 1 つを運用系オペレーティングシステムとし、他の 1 つを待機系オペレーティングシステムとして、運用系オペレーティングシステムに障害が生じたとき、待機系オペレーティングシステムに処理を継承させるクラスタシステムを 1 台の計算機内に構成した計算機システムの制御方法において、運用系オペレーティングシステムの障害を監視し、かつ、待機系オペレーティングシステムに処理を継承する動作を監視して、運用系オペレーティングシステムの稼働時、運用系オペレーティングシステムに待機系よりも多くの計算機資源を割り当てておき、運用系オペレーティングシステムに障害が発生して、待機系オペレーティングシステムが処理を開始した場合、待機系オペレーティングシステムに計算機資源を多く割り当てて、待機系オペレーティングシステムを運用系として動作させることを特徴とする計算機システムの制御方法。

【請求項 3】 複数のオペレーティングシステムを 1 台の計算機内で稼働させている計算機システムの制御方法において、前記オペレーティングシステムの稼働状態を監視して各 OS の負荷を監視すると共に、この負荷を分析し、高い負荷を持つオペレーティングシステムに関する負荷の原因を特定し、前記原因を解決するために必要な計算機資源を多く割り当てることを特徴とする計算機システムの制御方法。

【請求項4】 複数のオペレーティングシステムを1台の計算機内で稼働させている計算機システムの制御方法において、複数のオペレーティングシステムのそれぞれが扱う処理群を管理し、前期処理を他のオペレーティングシステムの処理と比較して優先的に処理を行う必要がある場合、優先的に処理を行う必要があるオペレーティングシステムに計算機資源を多く割り当てることを特徴とする計算機システムの制御方法。

【請求項5】 複数のオペレーティングシステムを1台の計算機内で稼働させている計算機システムの制御方法において、複数のオペレーティングシステムのそれぞれが扱う処理群と各処理を実行する稼働時間とを管理し、稼働時間によって計算機資源の変更内容と有効期間とを管理し、オペレーティングシステムの扱う処理と計算機資源の動的割り当てとを関連させて、各オペレーティングシステムに対する計算機資源の割り当てを行うことを特徴とする計算機システムの制御方法。

【請求項6】 複数のオペレーティングシステムのいずれにも割り当てられていない計算機資源を管理し、各オペレーティングシステムに割り当てられている計算機資源の使用率を管理し、この使用率に従って各オペレーティングシステムに、割り当てられていない計算機資源を割り当てることを特徴とする請求項3、4または5記載の計算機システムの制御方法。

【請求項7】 計算機システムに接続して計算機資源を使用する使用者を管理し、使用者の計算機資源の使用時間を管理し、使用者の要求により計算機資源の割り当てを変更し、使用者が設定した計算機資源の割り当てを行い、その割り当ての状態による課金を行うことを特徴とする請求項6記載の計算機システムの制御方法。

【請求項8】 複数のオペレーティングシステムを1台の計算機内で稼働させている計算機システムの制御方法において、複数のオペレーティングシステムに共通の主記憶領域を確保し、前記主記憶領域に通信用の形態を形成するデータをオペレーティングシステムが書き込みかつ読み出すことにより、前記オペレーティングシステム相互間で擬似的に通信媒体を形成することを特徴とする計算機システムの制御方法

【請求項 9】 複数のオペレーティングシステムを 1 台の計算機内で稼動させている計算機システムの制御方法において、前記各オペレーティングシステムが独立に稼動しており、共有資源を用いて、各オペレーティングシステムの動作を監視することを特徴とする計算機システムの制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、1 台の計算機上で複数のオペレーティングシステム（以下、OS という）が動作する計算機システム及び計算機システムの制御方法に係り、特に、複数の OS の稼動状態において、計算機資源を効率的に運用することを可能にした計算機システム及び計算機システムの制御方法に関する。

【0002】

【従来の技術】

計算機の信頼性向上、負荷分散のための制御に関する従来技術として、例えば、特開平 1 1 - 3 5 3 2 9 2 号公報等に記載された技術が知られている。この従来技術は、クラスタシステムと呼ばれるものであり、計算機上で動作するある OS が障害を起こした場合、別の OS が障害を起こした OS の処理を引き継いで処理を継続することを可能としたシステムである。これにより、この従来技術のシステムは、計算機上の OS に障害が発生した場合にも、計算機上で稼動しているプログラムを継続して実行することが可能となる。

【0003】

【発明が解決しようとする課題】

前述した従来技術による計算機システムであるクラスタシステムは、複数の OS を稼動させるために、クラスタを構成する OS の数と同数の計算機が必要である。また、従来のクラスタシステムは、運用系の計算機と待機系の計算機とを備えて、かつ、それぞれの計算機が独自に動作している。なお、運用系の計算機とは、通常の動作している計算機、待機系の計算機とは、運用系の計算機が障害となったとき、運用系で操作している処理を引き継ぐために必要な計算機である。

このため、従来技術によるクラスタシステムは、運用系の計算機に障害が起き

た場合、待機系の計算機に処理が引き継がれるが、待機系の計算機で独自に動作しているプロセスがある場合、運用系の計算機の全処理能力を引き継ぐことはできないという問題点を有している。また、従来技術のシステムは、運用系の計算機の処理能力を引き継ぐために待機系の動作を止めておくこととすると、待機系の計算機が使用されず、計算機資源を効率的に使用することができないという問題点を生じる。

【 0 0 0 4 】

例えば、2 台の計算機でクラスタシステムを構成した場合、このシステムは、1 台の計算機を運用系の計算機、もう 1 台の計算機を待機系の計算機として構成されるが、運用系の計算機が障害となったとき、その処理能力の全てを待機系の計算機が引き継ぐには、待機系の計算機が、運用系の計算機と同一の処理能力を持ち、かつ、運用系の計算機に障害が起ったとき、待機系の計算機で動作している処理を停止して、運用系の計算機から引き継いだ処理を行わなければならないことになる。また、待機系の計算機で動作している処理停止せずに、運用系の計算機から引き継いだ処理を行うためには、待機系の計算機として、運用系の計算機の処理能力以上の計算機を用いなければならないことになる。

【 0 0 0 5 】

すなわち、従来技術によるクラスタシステムは、通常の運用系の計算機 1 台の処理能力を達成するために、運用系の計算機以上の処理能力を持つ計算機を 2 台備えなければならないという問題点を有している。

【 0 0 0 6 】

本発明の目的は、前述した従来技術の問題点を解決し、計算機の信頼性の向上を図るためのクラスタシステムを構成しながら、計算機資源の使用の無駄を無くして効率的な運用を行うことのできる計算機システム及び計算機システムの制御方法を提供することにある。

【 0 0 0 7 】

【課題を解決するための手段】

本発明によれば前記目的は、複数の OS を 1 台の計算機システム内で動作させ、各 OS の状態を監視して、計算機資源を動的に変更、再配置を行うことにより

、計算機システム上のOSの状態により、計算機資源の効率的な運用を可能とすることにより達成される。例えば、1台の計算機システム上で動作する1つのOSを運用系OS、他の1つを待機系OSとしてクラスタシステムを構成し、運用系のOSに計算機資源を多く割り当て、運用系OSに障害が起った場合、待機系OSに運用系OSの処理を引き継がせ、そのとき、計算機資源を新たに運用系となる待機系OSに多く割り当てるようにする。これにより、計算機資源を効率よく運用することが可能となる。

【 0 0 0 8 】

また、前記目的は、計算機資源の変更、再配置の環境を用いて、OSの障害のみではなく、様々な状況を監視して、その状況に従い計算機資源を効率よく運用するようにすることにより達成される。例えば、あるOSの処理数の増加に伴うOSの負荷の増大に対して、そのOSに計算機資源を多く割り当てることにより、OSの処理能力を向上させ、負荷の減少につなげることが可能である。また、複数のOSの全ての負荷が増大した場合、待機させておいたCPUやメモリを加えて、処理能力を向上させ、終了すれば、その資源を解放することにより、計算機資源の効率的な運用を図ることもできる。

【 0 0 0 9 】

また、前記目的は、計算機資源の再配置に従い、計算機システムを使用する複数のユーザが、特定の状況で多くの資源を使いたい場合、それを計算機システムの管理側に登録し、計算機の資源の使用によって、使用ユーザに使用計算機資源の割合ごとに課金を行うようにすることにより達成される。これにより、計算機システムをサーバとして複数のユーザに展開することが可能となり、また、その使用の柔軟さと課金システムとを提供することができる。

【 0 0 1 0 】

また、前記目的は、1台の計算機システム内で複数のOSを稼働させる環境を作り出すOS管理制御部が全てのOSを管理できるようにし、複数のOSのために共有の主記憶領域を提供して、その共有の主記憶領域を用いて、外部入出力装置によらずに、主記憶経由でOS間通信を行わせるようにすることにより達成される。この通信手段を用いてネットワーク等と同等の通信用プロトコルを作り出

して、共有メモリの送受信を行うことにより、特別なプロトコルを採用することなく、擬似的に外部入出力装置を主記憶上に実現することが可能となる。

【0011】

【発明の実施の形態】

以下、本発明による計算機システム及び計算機システムの制御方法の実施形態を図面により詳細に説明する。

【0012】

図1は本発明の第1の実施形態による計算機システムの構成を示すブロック図、図2は計算機システム上で動作する各OSをクラスタ構成とした場合の概念的な構成を示すブロック図、図3は計算機の状態を管理するテーブルの構成を説明する図、図4はOSの障害時の計算機資源の変更処理の動作を説明するフローチャートである。図1～図3において、100は計算機システム、110はプロセッサ群、120は主記憶装置、130は外部入出力装置群、140は複数OS管理制御支援装置、150はバス、200は複数OS管理制御部、201、202は第1、第2OS、203、204はクラスタサービス、205、206は監視エージェント、207、208はアプリケーション、221、222は第1、第2OS用CPU群、230～232は端末である。

【0013】

本発明の第1の実施形態による計算機システム100は、図1に示すように、プロセッサ群110、主記憶装置120、外部入出力装置群130、複数OSを1台の計算機システム上で動作させるために設けられる複数OS管理制御支援装置140、及び、バス150により構成されている。複数OS管理制御支援装置140は、1台の計算機システムが複数のOSを実行することができるよう、計算機システム100の構成要素であるプロセッサ群110、主記憶装置120、外部装置群130の仲介をとる装置である。プロセッサ群110は、1つ以上のプロセッサの集合であり、プロセッサ(CPU)11a、11b、……、11nからなる。主記憶装置120は、複数の各OS-1～OS-n用に独立した主記憶領域12a、12b、……、12nと、複数OS管理制御部用の主記憶領域121と、使用されていない空き領域122とにより構成されている。外部装置

群 1 3 0 は、入力用のキーボード 1 3 1、出力用ディスプレイ 1 3 2、各 OS 用外部記憶装置 1 3 3 ~ 1 3 5、通信装置 1 3 6、1 3 7、その他の図示しないデバイスから構成されている。

【 0 0 1 4 】

前述のような構成を有する計算機システム 1 0 0 上で動作する各 OS をクラスタ構成とした場合の概念的な構成を示す図 2 の例は、OS - 1 と OS - 2 と（第 1 OS、第 2 OS ともいう）がクラスタ構成を取っている。そして、各 OS に割り当てた装置として、外部装置群 1 3 0 から、OS - 1 用の外部記憶装置 1 3 3、OS - 2 用の外部記憶装置 1 3 4、各 OS に共有のディスク装置 1 3 5、OS - 1 用の通信装置 1 3 6、1 3 7、OS - 2 用の通信装置 1 3 8、1 3 9 が備えられている。前述の OS - 1 用の通信装置 1 3 6 と OS - 2 用の通信装置 1 3 9 との間、及び、OS - 1 用の通信装置 1 3 7 と OS - 2 用の通信装置 1 3 8 との間は、それぞれケーブル 2 1 1、2 1 2 により接続されている。ケーブル 2 1 1 は、他の計算機に接続する外部接続用ケーブルであり、ケーブル 2 1 2 は、クラスタを構成する OS - 1 と OS - 2 との間のみを接続するクラスタ用接続ケーブルである。ケーブルは、2 種類あるが、これはクラスタの信頼性向上のためであり、ケーブル 2 1 2 を設けなくてもクラスタ構成を行うことは可能である。さらに、OS - 1、OS - 2 のそれぞれには、OS - 1 用 CPU 群 2 2 1、OS - 2 用 CPU 群 2 2 2 が割り当てられている。

【 0 0 1 5 】

前述において、主記憶装置 1 2 0 の複数 OS 管理制御部領域 1 2 1 には、複数 OS 管理制御部 2 0 0 が割り当てられており、複数 OS 管理制御部 2 0 0 は、計算機資源であるプロセッサ群 1 1 0、主記憶装置 1 2 0、外部入出力装置群 1 3 0 の構成を変更することが可能であるものとする。また、クラスタに接続される端末がある場合、それらの端末 2 3 0 ~ 2 3 2 は、ケーブル 2 1 1 に接続される。端末の 1 つ、例えば端末 2 3 0 は、OS - 1、OS - 2 の状態を管理し、その状態によって、計算機資源の変更の通知を、ケーブル 2 1 1、各 OS を介して行う管理端末としての役割を果たすことも可能である。

【 0 0 1 6 】

主記憶装置 1 2 0 内の OS - 1 領域 1 2 a、OS - 2 領域 1 2 b には、第 1 OS 2 0 1、第 2 OS 2 0 2 がそれぞれ割り当てられ、また、クラスタサービス 2 0 3、2 0 4、監視エージェント 2 0 5、2 0 6、各 OS で動作するアプリケーション 2 0 7、2 0 8 が格納されている。監視エージェント 2 0 5、2 0 6 は、計算機システムの状態を解析し、計算機資源を効率に運用するために、計算機システムの状態に応じて OS 管理制御部 2 0 0 に計算機資源の変更、再配置を要求するプログラムである。

【 0 0 1 7 】

第 1 及び第 2 OS 用 CPU 群 2 2 1、2 2 2 を構成する CPU の割り当て方法としては、次の 2 種類がある。1 つは、CPU 群 1 1 0 の中の CPU - 1 と CPU - 2 とを第 1 OS 用に用いる CPU 群 2 2 1 とし、CPU - 3 と CPU - 4 とを第 2 OS 用に用いる CPU 群 2 2 2 とするということのように、各 CPU を個々の OS でのみ使用する構成方法である。もう 1 つは、両方の CPU 群 2 2 0、2 2 1 に同じ CPU 1、CPU 2 を割り当てて、各 CPU が時間間隔でそれぞれ CPU を各 OS に振り分けるて使用する構成方法である。本発明は、これらのどの方法用いて各 OS 用の CPU 群を構成してもよい。そして、本発明は、クラスタシステムを構成する複数の OS に対して、その OS の状況に応じて、CPU、メモリ等の計算機資源を変更可能に割り振ることを可能にしており、例えば、2 つの OS がクラスタシステムを構成し、その一方が運用系、他方が待機系として動作する場合、運用系の OS に対して多くの計算機資源を割り当てるようにしている。この計算機資源の各 OS への割り当ては、計算機資源管理テーブル 3 0 0 により管理される。

【 0 0 1 8 】

クラスタ構成となっている OS - 1 と OS - 2 とに対する計算機資源の割り当てを管理する計算機資源管理テーブル 3 0 0 は、図 3 にその例を示すように構成されており、主記憶装置 1 2 0 内の図示しない領域に格納されて、複数 OS 管理制御支援装置 1 4 0 が管理可能であればよい。なお、以後の説明の全てのテーブルについても同様である。計算機資源管理テーブル 3 0 0 は、計算機資源割り当てテーブル 3 1 0 と OS 状態テーブル 3 2 0 との 2 つのテーブルにより構成され

ている。このテーブル 3 0 0 を管理する複数 O S 管理制御支援装置 1 4 0 内に設けられる図示しない各資源対応の処理実行部は、このテーブルの値を基準に、計算機の状態と計算機資源の状態とを把握して、計算機資源の割り当ての変更を行う。計算機資源割り当てテーブル 3 1 0 には、計算資源名 3 1 1 と、運用系の計算機に対する計算機資源サービス比 3 1 2 と、待機系の計算機に対する計算機資源サービス比 3 1 3 とが格納される。これにより、計算機資源とクラスタシステムにおける計算機の役割との関連が設定される。O S 状態テーブル 3 2 0 には、O S 名 3 2 1 と、各 O S の動作状態を示す情報 3 2 2 と、各 O S が運用系か待機系かを示す情報 3 2 3 と、各 O S に対する資源（C P U）のサービス比率 3 2 4 が格納される。この O S 状態テーブル 3 2 0 により、各 O S の動作状況を把握することが可能である。

【 0 0 1 9 】

前述したような構成を持つ計算機資源管理テーブル 3 0 0 は、クラスタ構成、前述の各資源対応の処理実行部の役割の割り当てによって、テーブルを管理する処理実行部により変更される。図 3 に示すテーブルにおいては、運用系の O S - 1 に対して、C P U 資源の 9 5 % が割り当てられ、残りの 5 % が、待機系の O S - 2 に対して割り当てられていることを示している。また、図 3 には、主記憶装置の割り当て量について記載されていないが、主記憶装置の割り当て量も、運用系の O S - 1 に対して多くが割り当てられる。なお、主記憶の割り当ては、サービス比率でなくアドレス範囲で行ってもよい。また、計算機資源割り当てテーブル 3 1 0 には、C P U と主記憶とを、資源名として登録しているが、外部記憶装置等についても、その使用台数等について登録しておくことができる。

【 0 0 2 0 】

次に、図 4 に示すフローを参照して、O S の障害時の計算機資源の変更処理の動作を説明する。この処理は、前述した計算機資源テーブル 3 0 0 を各監視エージェント 2 0 5、2 0 6 が管理し、クラスタサービス 2 0 3、2 0 4 と、O S 管理制御部 1 2 1 と、監視エージェント 2 0 5、2 0 6 とが連携して、運用系の O S である O S - 1 に障害が起きた場合に計算機資源の変更を行う場合を例とした処理である。

【0021】

(1) 各監視エージェント205、206は、定期的にクラスタサービス203、204やOS管理制御部121との間で通信を行い、クラスタサービスまたはOS管理制御部から、自監視エージェントが動作しているOS以外のOSに障害が生じたか否かの情報を獲得する(ステップ400)。

【0022】

(2) 各監視エージェント205、206は、自監視エージェントが動作しているOS以外のOSであって、運用系のOSに障害が生じたか否かを判定し、障害が生じていなければ、何の処理も行わず、ここでの処理を終了する(ステップ401)。

【0023】

(3) 運用系のOSであるOS-1に障害を生じたとすると、ステップ401での判定で、待機系OSで動作している監視エージェント206が運用系のOSであるOS-1に障害を生じたことを検出することになる。運用系のOSであるOS-1に障害を生じたことを検出すると、待機系OS-2の監視エージェント206は、計算機資源割り当てテーブル310より、障害が起ったとき計算機資源をどのように変更するかを獲得する(ステップ402)。

【0024】

(4) ここで説明している例は、待機系OSで動作している監視エージェント206が運用系OSでの障害を検出したとしているので、運用系OS-1で障害が生じたとき、待機系OSは、運用系の処理を引き継ぎ、引き継いだ後、運用系OSとして動作することになる。このため、計算機資源として、今まで待機系OS用のCPUの割り当て率であったものを、運用系の値に変更することになり、監視エージェント206は、この計算機資源の変更を実際にCPUの動作を管理するOS管理制御部200に通知する(ステップ403)。

【0025】

(5) そして、OS管理制御部が実際の処理を行った後、OS状態テーブル320を変更する。ここでの変更は、OS状態テーブル320の動作状態と、系の種類と、CPU利用率との変更であり、障害発生後、今までの運用系OSが待機系

OSに、今までの待機系OSが運用系OSとなり、CPU利用率も、この系の変更に従ったテーブルに設定されている値に変更される。主記憶の利用率についても同様に変更される（ステップ404）。

【0026】

（6）そして、障害が生じて待機系とされたOSが復帰した場合に備えて、復帰したOSで再び稼動する監視エージェントに対して、今現在の情報、すなわち、復帰したOSが待機系として動作することを通達しておき処理を終了する（ステップ405）。

【0027】

図4により説明した例は、計算機状態テーブル300を監視エージェントが監視するとたが、計算機状態テーブル300の監視を監視エージェントはでなく、管理端末230で管理することもできる。この場合、各OSの状態を集中的に管理端末230により管理することができるので、ステップ405の処理を行う必要がなくなる。また、管理端末230に、クラスタサービスのクライアントサービスに接続して、各クラスタの状況を把握することも可能である。この場合、管理端末230は、ステップ400の処理で、定期的に各OSの稼動状況をクラスタサービスに問い合わせればよい。また、前述において、待機系OSと運用系OSとは、基本的には同一のものが使用されるが、同一のアプリケーションの処理を実行することができるものであれば、異なる種類のOSであってもよい。

【0028】

前述したように、本発明の第1の実施形態は、CPUの資源を、運用系として稼動するOSに多くの利用率で割り当てておき、運用系のOSに障害が発生したとき、その処理を引き継いで新たに運用系となるOSに、CPUの資源を多くの利用率で再割り当てすることができ、これにより、計算機資源の有効活用を実現することが可能である。

【0029】

前述した本発明の第1の実施形態は、OSの運用形態に従って、すなわち、OSが運用系であるか待機系であるかにより計算機資源の割り当て比率を決めて、運用形態が変更になったとき、それに従って計算機資源の割り当てを変更すると

して説明したが、本発明は、各OS上の処理の負荷の大きさや特定の時間に、計算機資源の割り当てを変更して、計算機資源の効率的な運用を図るようにすることもできる。以下、本発明の第2の実施形態を説明する。

【0030】

図5は本発明の第2の実施形態に必要な各種のテーブルの構成を説明する図、図6はOSに負荷がかかった場合の計算機資源割合変更対策を設定した計算機資源変更テーブルの構成を説明する図、図7は計算機資源の負荷を表す計算機資源変更状態テーブルの構成を説明する図、図8は各OSが計算機資源の変更を行う処理手順について説明するフローチャートである。

【0031】

図5に示す本発明の第2の実施形態に必要な各種のテーブルは、計算機システム全体の資源と、それに対する各OSの資源利用設定とを表すもので、主記憶装置のデータ領域500に格納されている。このデータ領域500には、CPU使用テーブル510、主記憶使用テーブル520、外部装置使用テーブル530、物理CPU設定テーブル540、OS設定テーブル550が格納されている。

【0032】

CPU使用テーブル510には、項目511として、稼動しているCPU数と稼動していないCPU数とが登録され、これらに対するCPU数512の値が登録されており、資源変更のときに、これらのデータが用いられる。主記憶使用テーブル520には、項目521として、使用されているメモリ容量と使用されていないメモリ容量とが登録され、これらに対する値522が登録されている。また、外部装置使用テーブル530には、項目531として、各外部装置であるI/Oが登録され、それらが使用されているか否かを示す情報532が登録されている。物理CPU設定テーブル540は、CPU群110の中で、独立して用いるCPUの設定状況を示している。このテーブル540には、項目541として、分割のための各OSのIDが登録され、それらの各OSに対して、それらのIDを持つOS相互間での優先度542、初期に分割して割り当てたCPU数である初期個数543、このテーブルを定期的に更新する監視エージェントにより実際に計算機資源を変更したときに更新された現在のCPU数を表す現在個数54

4 が登録されている。優先度 5 4 2 は、OS がさらに多くの CPU を要求した場合に、その変更を行う場合の優先度である。OS 設定テーブル 5 5 0 は、各 OS の初期設定を登録するテーブルであり、このテーブル 5 5 0 には、各 OS の OS 名 5 5 1 に対して、各 OS 間の実行優先度 5 5 2、OS の実行系の種類 5 5 3、物理 CPU 設定テーブルの ID を表す ID 5 5 4、CPU サービス比 5 5 5、主記憶の使用容量 5 5 6、使用外部装置 5 5 6、5 5 7 がそれぞれ登録されている。

【 0 0 3 3 】

前述において、使用外部装置は、計算機システム 1 0 0 の接続状況によりその個数が増減される。また、OS の実行系の種類 5 5 3 とは、継続的に処理を行うトランザクション系と、蓄積してある処理を集中して行うバッチ系とである。本発明の第 2 の実施形態は、この処理系の性質の違いを考慮して、計算機資源を変更することにより、無駄のない計算機資源の割り当ての制御を行う。

【 0 0 3 4 】

図 6 に示す OS に負荷がかかった場合の計算機資源割合変更対策を設定した計算機資源変更テーブル 6 0 0 には、OS 名 6 0 1 と、その OS に対応する各変更ルールの ID 6 0 2 とが登録され、さらに、これらのそれぞれに対して次に説明するような情報が登録される。待機フラグ 6 0 3 は、各ルールを示す ID 6 0 2 の条件に対して、実際に計算機資源の変更が可能でない状況がある場合、計算機資源の変更を再度行うか否かを示すフラグであり、この待機フラグ 6 0 3 には数値が入れられる。このフラグ 6 0 3 は、定期的に監視を行うエージェントが、後述する図 8 のフローによる処理を実行するたびに、計算機資源の変更が可能であるか否かを調査するために使用される。その調査回数が待機フラグの値である。OS 名 6 0 1 に対して、監視エージェント 2 0 5、2 0 6 は、OS の負荷を、計算機資源変更条件 6 0 4 に基づいて調査する。計算機資源変更条件 6 0 4 には、各計算機資源の CPU 6 0 6、主記憶 6 0 7、I/O（外部装置）6 0 8 に関する閾値データが格納されている。そして、監視エージェント 2 0 5、2 0 6 は、調査の結果、登録条件を満たせば、計算機資源変更設定 6 0 5 の中の各計算機資源である CPU 6 0 9、主記憶 6 1 0、I/O 6 1 1 の値に基づいて設定要求を

行う。また、OS共通の設定がある場合、各OSがOS共通のルールを選択する。

【0035】

図7に示す各OSの計算機資源の負荷を表す計算機資源変更状態テーブル700は、OS名701と各OSの状態と格納しており、各OSの状態として、各OSに割り当てられている計算機資源の負荷が格納され、監視エージェントは、この負荷の値を考慮して計算機資源変更の判定基準とする。また、変更後の資源状態も管理し、今後値を戻すためのデフォルト値と共に管理される。各OS名701に対する現在の算機資源として、CPU702、メモリ704、I/O706、I/O2(708)が登録され、そのそれぞれについての負荷である、CPU負荷703、メモリ負荷705、I/O負荷707、I/O2負荷709が格納される。この計算機資源の状態は、計算機資源変更ルールを用いるか否かの判断基準となる。

【0036】

次に、図8に示すフローを参照して、各OSが計算機資源の変更を行う処理手順について説明する。この処理は、監視エージェントがOSと連携して実行する処理である。

【0037】

(1) 監視エージェントは、定期的に各OSと連携してOSの稼動状態を調査し、その状態をOS状態テーブルである計算機資源の負荷を表す計算機資源変更状態テーブル700に格納し、計算機資源変更テーブル600の情報に基づいて、連携しているOSの負荷が高負荷となっているか否かをチェックする(ステップ800、801)。

【0038】

(2) ステップ801でのチェックで、そのOSの負荷が高負荷となっていなかった場合、OSに割り当てられている資源がデフォルト値と同一か否かをチェックし、デフォルト値と同一であった場合、計算機資源を元の状態に戻す必要がないので、何も行わずに処理を終了する(ステップ805)。

【0039】

(3) ステップ805のチェックで、OSに割り当てられている資源がデフォルト値と同一でなかった場合、資源の状態を以前の状態に戻すことができるか否かをチェックし、資源の状態を以前の状態に戻すことができる状態ではない場合、何も行わずに処理を終了する(ステップ806)。

【0040】

(4) ステップ801のチェックで、そのOSの負荷が高負荷となっていた場合、そのOSの優先度が1番であるか否かをOS設定テーブル550により調べ、優先度が1番でない場合、計算機資源変更テーブル700から他のOSの負荷状況を獲得し、他のOSの負荷が大きくなり、現在計算機資源変更が可能か否かを調べる(ステップ802、803)。

【0041】

(5) ステップ802のチェックで、そのOSの優先度が1番であった場合、または、ステップ803のチェックで、他のOSの負荷が大きくなり、現在計算機資源の変更が可能であると判定された場合、あるいは、ステップ806のチェックで、資源の状態を以前の状態に戻すことができる状態であると判定した場合、計算機資源変更テーブル600から資源変更のデータを獲得する(ステップ807)。

【0042】

(6) ステップ803のチェックで、他のOSの負荷が大きく、現在計算機資源の変更を行うことが不可能であると判定した場合、連携しているOS及び他のOSの待機フラグ603が設定されているか否かをチェックし、いずれかのOSの待機フラグが設定されていれば、連携しているOSの待機フラグの設定を変更(+1)して、ここでの処理を終了する(ステップ804)。

【0043】

(7) ステップ804のチェックで、待機フラグ603が設定されていなかった場合、計算機資源変更テーブル600から資源変更のデータを獲得し、それに見合う計算機資源の追加資源量を獲得する(ステップ808)。

【0044】

(8) 前述したステップ807あるいはステップ808の処理後、実際に計算機

資源を変更可能なOS連携制御部に、計算機資源の変更要求を出し、計算機資源の変更処理の終了後、状態テーブル700を変更する（ステップ809、810）。

【0045】

本発明の第2の実施形態によれば、前述したような処理を行うことによって、OSの負荷の状態を監視し、その負荷の状態に従って、計算機資源の変更要求を行って、効率的に資源の運用を行うことが可能に、OSに対して計算機資源の割り当てを行うことができる。

【0046】

前述した本発明の第2の実施形態は、OSの負荷の状態に応じて、OSに割り当てる計算機資源を動的に変更するものであったが、本発明は、複数のOSのそれぞれが扱う処理群を管理し、この処理を他のOSの処理と比較して優先的に処理を行う必要がある場合、優先的に処理を行う必要があるOSに計算機資源を多く割り当てるようにすることもできる。また、本発明は、複数のOSのいずれにも割り当てられていない計算機資源を管理しておき、各OSに割り当てられている計算機資源の使用率を監視して、この使用率が高くなったOSに、割り当てられていない計算機資源を割り当てるようにすることができる。

【0047】

また、前述した本発明の第2の実施形態は、バッチ処理系に対する計算機資源の割り当てに適用することにより、集中的なバッチ処理を行うことが可能となり、バッチ処理を効率的に実行することができる。以下、この例について説明する。

【0048】

図9はバッチ処理系に対する計算機資源変更用のテーブルの構成を説明する図、図10はバッチ処理系の計算機資源変更の処理動作を説明するフローチャートである。ここで説明する例は、バッチ処理系の計算機資源変更を、そのバッチ処理が行われる毎に、他の計算機資源に余裕があり余っていれば、その計算機資源をバッチ処理系割り当てて、そのバッチ処理を集中的に行わせることを可能にしたものである。

【0049】

図9に示す計算機資源変更テーブル900には、バッチ処理のジョブ名901に関して、ジョブを実行するOS名902、ジョブの開始時間903、終了時間904、及び、開始、終了の時間の間での各計算機資源の変更条件906が計算機資源変更テーブル600の場合と同様に登録される。また、計算機資源変更テーブル600の場合と同様な待機フラグ905も設定されている。

【0050】

次に、図10に示すフローを参照して、バッチ処理系の計算機資源変更の処理動作について説明する。

【0051】

(1) 監視エージェント205、206は、テーブル900によりジョブの開始時間を監視し、現時間がジョブの開始時間になったか否かを調べ、ジョブ開始時間でなければ、何の処理も行わずに、ここでの処理を終了する(ステップ1000、1001)。

【0052】

(2) ステップ1001のチェックで、現時間がジョブの開始時間であれば、図8により説明した場合と同様に、バッチ系のOSの優先度が1番となっているか否かを調べ、優先度が1番となっている場合、それなりの計算機資源が割り当てられているので、何の処理も行わずに、ここでの処理を終了する(ステップ1002)。

【0053】

(3) ステップ1002のチェックで、優先度が1番でない場合、他のOSの負荷を調べ、他のOSに大きな負荷が掛かっている場合、計算機資源の割り当て不可能として、何の処理も行わずに、ここでの処理を終了する(ステップ1003)。

【0054】

(4) ステップ1003のチェックで、他のOSに負荷が掛かっていなかった場合、待機フラグを調べて他に計算機資源変更を要求して待ちになっているOSがあるか否かを調べて、待機フラグが設定されているOSがあれば、待機フラグ9

03を設定して処理を終了する（ステップ1004）。

【0055】

（5）ステップ1004のチェックで、待機フラグが設定されていなかった場合、計算機資源の変更が可能であると判定し、計算機資源変更テーブル900から、変更条件を獲得し、それを実際に計算機資源を変更するOS管理制御部240に通達する。そして、資源の変更が終了すれば、変更後OS状態テーブルに状況を書き込んで処理終了する（ステップ1005～1007）。

【0056】

前述した処理動作の後、バッチ処理系は、変更された計算機資源を使用してバッチ処理を実行する。そして、監視エージェント205、206は、その処理の終了後、または、計算機資源変更テーブル900のジョブ終了時間904になった場合、計算機資源の状態を変更前の元の状況に戻す。

【0057】

前述したように、計算機の状態を監視エージェントにより管理し、適宜計算機資源を変更することにより、計算機資源を複数のOSに対して有効に割り当てることが可能である。

【0058】

一般に、マルチプロセッサ環境の計算機をサーバ機として用いる場合、複数のユーザがサーバ計算機にアクセスする。その際、アクセスする計算機の処理能力を保証するため、計算機資源が分割してユーザ毎に割り当てられる。このようにサーバ計算機に対して本発明による計算機資源を変更する手段を与え、その変更の状況に応じて課金を行うようにすることが可能である。この課金方法によって、ユーザに柔軟なサーバ機の利用を行わせることができる。以下、計算機資源変更技術を用いた課金方法を本発明の第3の実施形態として説明する。

【0059】

図11は本発明の第3の実施形態に必要なユーザ課金テーブルとユーザ登録テーブルとを持つユーザ設定領域の構成を説明する図、図12は課金計算に必要な各種の課金基準テーブルを持つ計算機資源課金基準領域の構成を説明する図、図13は計算機資源の利用を基準とした課金処理の手順を説明するフローチャート

である。

【0060】

図11に示すユーザ設定領域1100は、主記憶装置ないに格納され、ユーザ課金テーブル1110とユーザ登録テーブル1120とにより構成される。ユーザ課金テーブル1110は、計算機資源を変更するユーザからの要求に応じて、そのテーブルの内容が更新されていく。ユーザ課金テーブル1110には、各ユーザを表すユーザID1111に対して、使用加算時間1112、CPU課金状況1113、1114、主記憶課金状況1115、1116、外部装置（I/O）課金状況1117、1118、合計課金状況1119に各課金値が更新可能に設定される。ユーザ登録テーブル1120には、ユーザが初期に登録したデータが格納されており、各ユーザを表すユーザID1121に対して、このテーブルの各項目である使用開始時間1122、使用終了時間1123、CPU登録1124、主記憶登録1125、外部装置（I/O）登録1126が初期設定として登録され、また、それに対する標準料金1127も計算されて登録されている。

【0061】

本発明の第3の実施形態は、前述のテーブル1110、1120への設定情報を基準として課金を行っていく。前述のテーブルにおいて、テーブル内の丸印は、課金がかからないことを表している。すなわち、ユーザ1がCPUを1個使用しても、標準料金以外に料金がかからないことを意味する。そして、この基準以上の計算機資源をユーザが使用した場合、それに対する課金が行われ標準料金に加算されることになる。

【0062】

図12に示す計算機資源課金基準領域1200には、それぞれの条件に基づいて課金設定を行い、課金を行うためのテーブルであるCPU課金基準テーブル1210、メモリ課金基準テーブル1230、外部装置課金基準テーブル1250が設けられており、これらのテーブルには課金の計算の基準となる値が格納されている。図12における例において、各テーブルの項目は、各計算機資源の使用時間と使用量との関係で決められる課金基準である。

【0063】

次に、図 1 3 に示すフローを参照して、計算機資源の利用を基準とした課金手順について説明する。

【 0 0 6 4 】

(1) まず、ユーザの使用確認を行い、そのユーザの条件である使用ユーザ指定処理の実行要求を獲得する。その後、計算機資源の要求があるか否かを判定し、もしなければ、資源の変更を行わず、他のユーザ等からの要求の処理に移る（ステップ 1 3 0 0 ～ 1 3 0 2）。

【 0 0 6 5 】

(2) ステップ 1 3 0 2 の判定で、ユーザが資源変更の要求を行っていれば、その条件に基づいて OS 管理制御部 2 0 0 に処理の要求を行い、計算機資源の変更が行われた後、計算機資源状態テーブル 7 0 0 を変更し、課金テーブルにその条件を登録する（ステップ 1 3 0 3、1 3 0 4）。

【 0 0 6 6 】

(3) そして、その課金状態が終了すると、計算機資源の状態を元の状態に戻し、課金テーブルに、その使用時間と課金情報とを格納して処理を終了する（ステップ 1 3 0 5）。

【 0 0 6 7 】

前述したような本発明の第 3 の実施形態によれば、ユーザは、柔軟に計算機環境を変更することができ、必要のあるときのみ、機能の高い計算機システムの提供を受け、必要がない場合、最低限の機能の計算機システムの提供を受けて、コスト的に柔軟に対応することが可能となる。

【 0 0 6 8 】

図 1 4 は本発明の第 4 の実施形態による計算機システム上で動作する各 OS をクラスタ構成とした場合の概念的な構成を示すブロック図、図 1 5 は共有メモリ通信の送信処理手順を説明するフローチャート、図 1 6 は共有メモリ通信の受信処理手順を説明するフローチャートであり、以下、本発明の第 4 の実施形態について説明する。

【 0 0 6 9 】

本発明の第 4 の実施形態は、OS 管理制御部 2 0 0 に各 OS に共通の共有メモ

リ 1 4 0 0 をおき、その共有メモリに通信用のフォーマットを作成して書き込み読み出しを行うことにより、通信ケーブルを必要としない擬似的な通信網を作成した共有メモリ通信を行うものである。そして、この実施形態は、図 1 4 に示すように、図 2 に示した構成において、OS 管理制御部に OS 間の共有メモリ 1 4 0 0 を付属させたものであり、クラスタ構成を組むときの内部通信網として共有メモリ通信をおこなうことも可能である。また、この実施形態は、図 2 に示した例で説明したケーブル 2 1 2 を不要として、独自の内部通信網を確立することができる。また、共有メモリ通信は、各 OS 領域に設けられる共有メモリ通信ドライバ 1 4 0 1、1 4 0 2 により行われる。

【0070】

共有メモリ通信の送信手順は、図 1 5 に示すフローに従って、次のように行われる。

【0071】

(1) 共有メモリ通信ドライバは、OS 上のプログラムから送信要求を受け取ると、送信データを通信用のフォーマットに変換する。例えば、通信として、TCP/IP によるイーサネットの通信を想定しているのであれば、そのフォーマットを生成する (ステップ 1 5 0 0、1 5 0 1)。

【0072】

(2) そして、共有メモリ 1 4 0 0 への書き込み要求を OS 管理制御部 2 0 0 に要求する (ステップ 1 5 0 2)。

【0073】

(3) OS 管理制御部 2 0 0 は、共有メモリ 1 4 0 0 に空きがあるか否かを判定し、空きがなければ、一度処理を終わり、共有メモリ 1 4 0 0 に空きができるまで待つ (ステップ 1 5 0 3)。

【0074】

(4) ステップ 1 5 0 3 判定で、共有メモリ 1 4 0 0 に空きがあれば、送信データを共有メモリに書き込む (ステップ 1 5 0 4)。

【0075】

前述した処理において、実際に共有メモリに送信データを書き込むのは OS 管

理制御部 2 0 0 であるので、通信ドライバ 1 4 0 1、1 4 0 2 は、OS 管理制御部に対して書き込みデータを伝えることになる。

【0 0 7 6】

共有メモリ通信の受信手順は、図 1 6 に示すフローに従って、次のように行われる。なお、OS は、一定時間毎に共有メモリ通信ドライバに受信要求を行い、受信すべきデータがある場合に、受信データを受け取るようにすることができ、あるいは、送信側となった OS の共有メモリ通信ドライバが、受信側に何らかの方法でデータの送信を知らせるようにすることもできる。

【0 0 7 7】

(1) 共有メモリ通信ドライバは、OS 上のプログラムから受信要求を受け取ると、共有メモリ 1 4 0 0 に受信用のデータがあるか否かをチェックする。すなわち、共有メモリ 1 4 0 0 に自分が受信するデータが書き込まれているか否かを調べる。そして、もし受信データがなければ、ここでの処理を終了する（ステップ 1 6 0 0、1 6 0 1）。

【0 0 7 8】

(2) ステップ 1 6 0 1 のチェックで、共有メモリに受信データがあれば、その受信データを共有メモリから読み出し、受信データの通信フォーマットを解析して、受信データを取り出す（ステップ 1 6 0 2、1 6 0 3）。

【0 0 7 9】

前述した例によれば、共有メモリを用いた送受信方法により、通信用のケーブルを備えることなく、共通するメモリエリアを用いて擬似的に通信ケーブルを実現することが可能である。この方法は、クラスタにおけるネットワーク通信等にも利用することができる。また、メモリ内に不揮発的に扱えるメモリエリアあれば、そこを擬似的な外部記憶装置にみたてて、その間を、SCSI インターフェイスを用いてアクセスすることにより、擬似的な SCSI 通信を行うことも可能である。

【0 0 8 0】

【発明の効果】

以上説明したように本発明によれば、1 台の計算機に複数の OS を稼働させた

計算機システムであって、各OSに対する計算機資源の変更、再配置を行うことが可能な環境の中で、クラスタシステムを構成することにより、各OSの運用状況を監視して、運用系OSの障害時に計算機資源を別の正常なOSに多く割り当てて、それを運用系とすることにより、障害に関わらず、計算機システムの処理能力を変えことなく計算機システムを稼働させることができる。

【0081】

また、本発明によれば、OSの負荷を監視して、その負荷に応じて、計算機資源を割り当てることが可能であるので、OS間の処理能力を必要に応じて調整することが可能である。

【図面の簡単な説明】

【図1】

本発明の第1の実施形態による計算機システムの構成を示すブロック図である。

【図2】

計算機システム上で動作する各OSをクラスタ構成とした場合の概念的な構成を示すブロック図である。

【図3】

計算機の状態を管理するテーブルの構成を説明する図である。

【図4】

OSの障害時の計算機資源の変更処理の動作を説明するフローチャートである。

【図5】

本発明の第2の実施形態に必要な各種のテーブルの構成を説明する図である。

【図6】

OSに負荷がかかった場合の計算機資源割合変更対策を設定した計算機資源変更テーブルの構成を説明する図である。

【図7】

計算機資源の負荷を表す計算機資源変更状態テーブルの構成を説明する図である。

【図 8】

各 OS が計算機資源の変更を行う処理手順について説明するフローチャートである。

【図 9】

バッチ処理系に対する計算機資源変更用のテーブルの構成を説明する図である。

【図 1 0】

バッチ処理系の計算機資源変更の処理動作を説明するフローチャートである。

【図 1 1】

本発明の第 3 の実施形態に必要なユーザ課金テーブルとユーザ登録テーブルとを持つユーザ設定領域の構成を説明する図である。

【図 1 2】

課金計算に必要な各種の課金基準テーブルを持つ計算機資源課金基準領域の構成を説明する図である。

【図 1 3】

計算機資源の利用を基準とした課金処理の手順を説明するフローチャートである。

【図 1 4】

本発明の第 4 の実施形態による計算機システム上で動作する各 OS をクラスタ構成とした場合の概念的な構成を示すブロック図である。

【図 1 5】

共有メモリ通信の送信処理手順を説明するフローチャートである。

【図 1 6】

共有メモリ通信の受信処理手順を説明するフローチャートである。

【符合の説明】

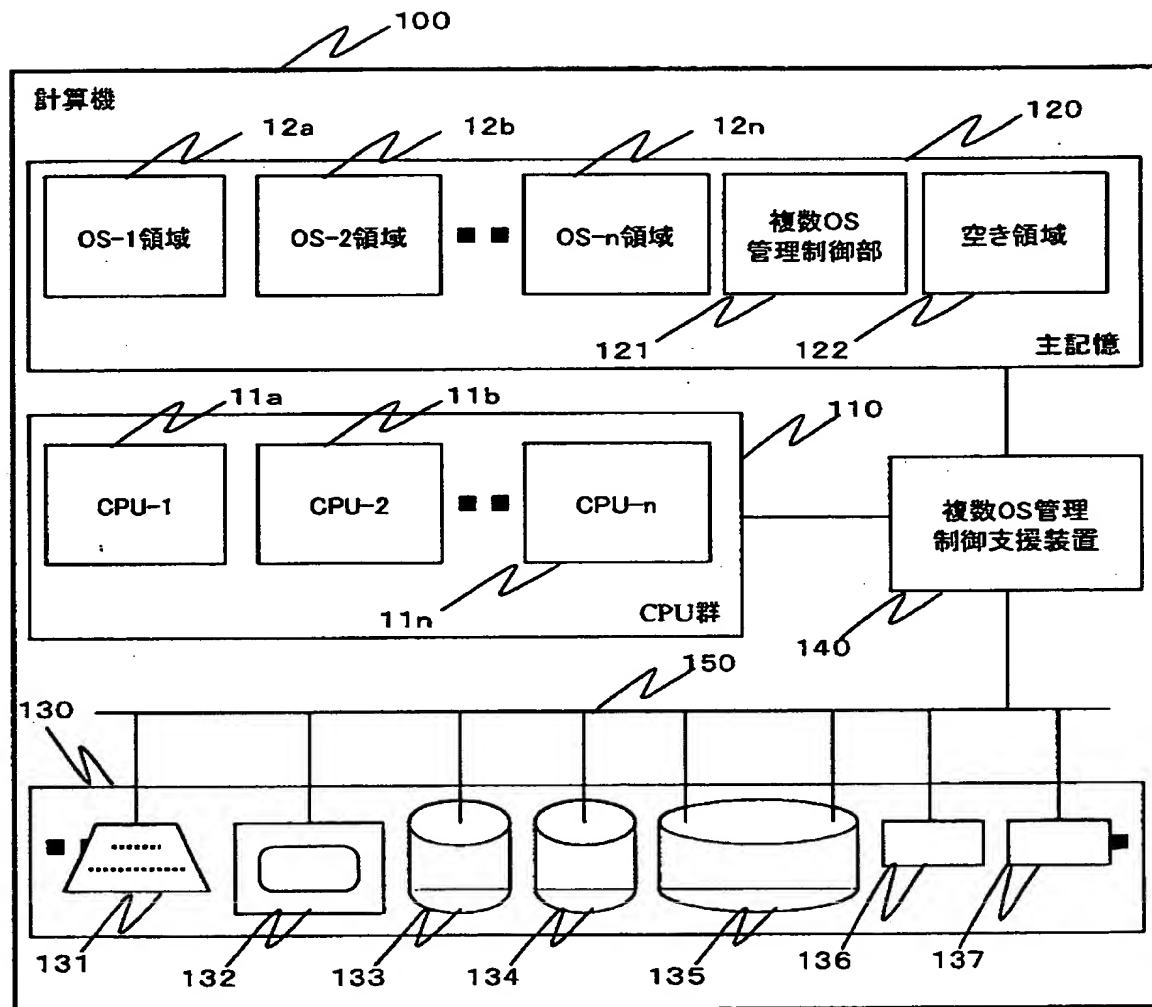
- 1 0 0 計算機システム
- 1 1 0 プロセッサ
- 1 2 0 主記憶装置
- 1 3 0 外部入出力装置群

- 1 4 0 複数 O S 管理制御支援装置
- 1 5 0 バス
- 2 0 0 複数 O S 管理制御部
- 2 0 1、2 0 2 第 1、第 2 O S
- 2 0 3、2 0 4 クラスタサービス
- 2 0 5、2 0 6 監視エージェント
- 2 0 7、2 0 8 アプリケーション
- 2 2 1、2 2 2 第 1、第 2 O S 用 C P U 群
- 2 3 0 ~ 2 3 2 端末

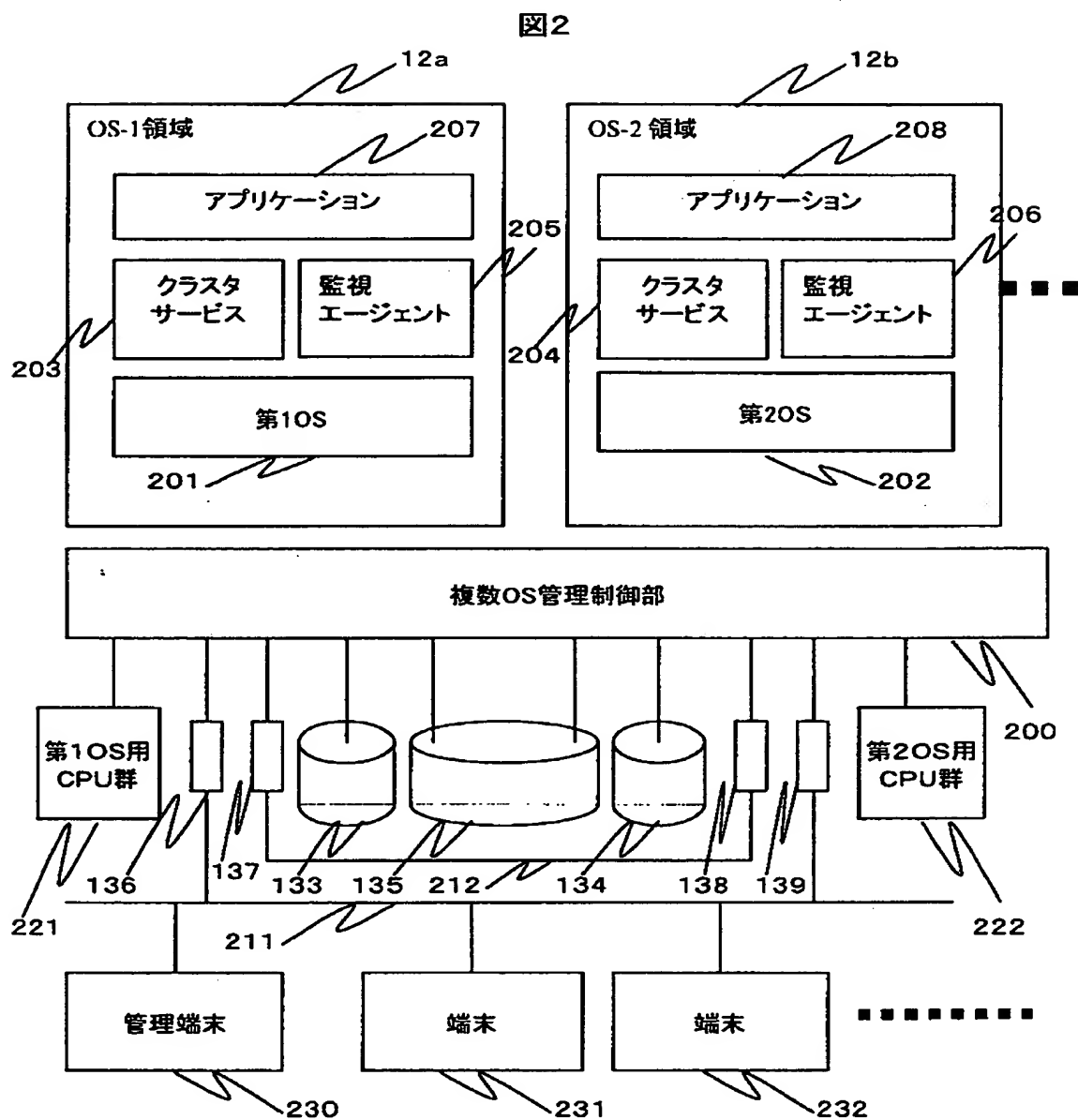
【書類名】 図面

【図1】

図1



【図2】



【図 3】

図 3

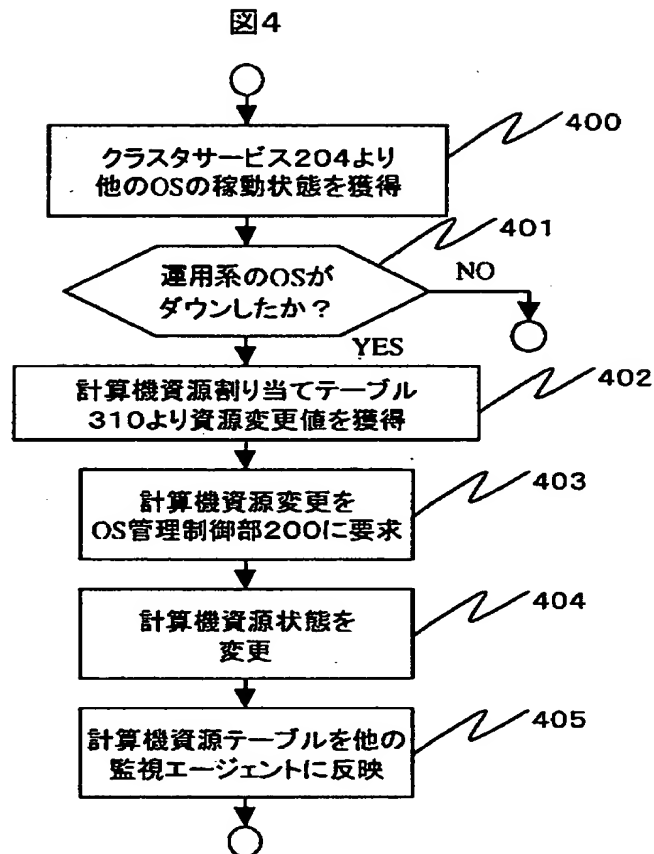
300			
311	312	313	310
計算機 資源名	運用系の計算機 資源サービス比	待機系の計算機 資源サービス比	
CPU	95%	5%	
主記憶	-	-	

321	322	323	324
OS 名	動作状態	運用系, 待機系	CPU
OS 1	稼動	運用系	95%
OS 2	稼動	待機系	5%

320

300: 計算機状態テーブル
 310: 計算機資源割り当てテーブル
 320: OS 状態テーブル

【図4】



【図 5】

図5

500

510 511 512 520 521 522

項目	値
使用 CPU 数	
空き CPU 数	

項目	値
使用メモリ数	
空きメモリ数	

530 531 532

項目	使用 / 未使用
I/O 1	
I/O 2	
I/O n	

540 541 542 543 544

ID	優先度	初期個数	現在個数
1	1	2	2
2	2	1	1

550 551 552 553 554 555 556 557 558

OS 名	優先度	種類	ID	CPU	主記憶	I/O 1	I/O 2
OS 1							
OS 2							
OS n							

500: 計算機資源設定領域
 510: CPU使用テーブル
 520: 主記憶使用テーブル
 530: 外部装置使用テーブル
 540: 物理CPU設定テーブル
 550: OS設定テーブル

【図 6】

図 6

OS 名	I D	待 機 フ ラ グ	計算機資源変更条件			計算機資源変更設定		
			CPU	主記憶	I/O	CPU	主記憶	I/O
OS1								
OS2								
Osn								
OS 共通								

600:計算機資源変更テーブル1

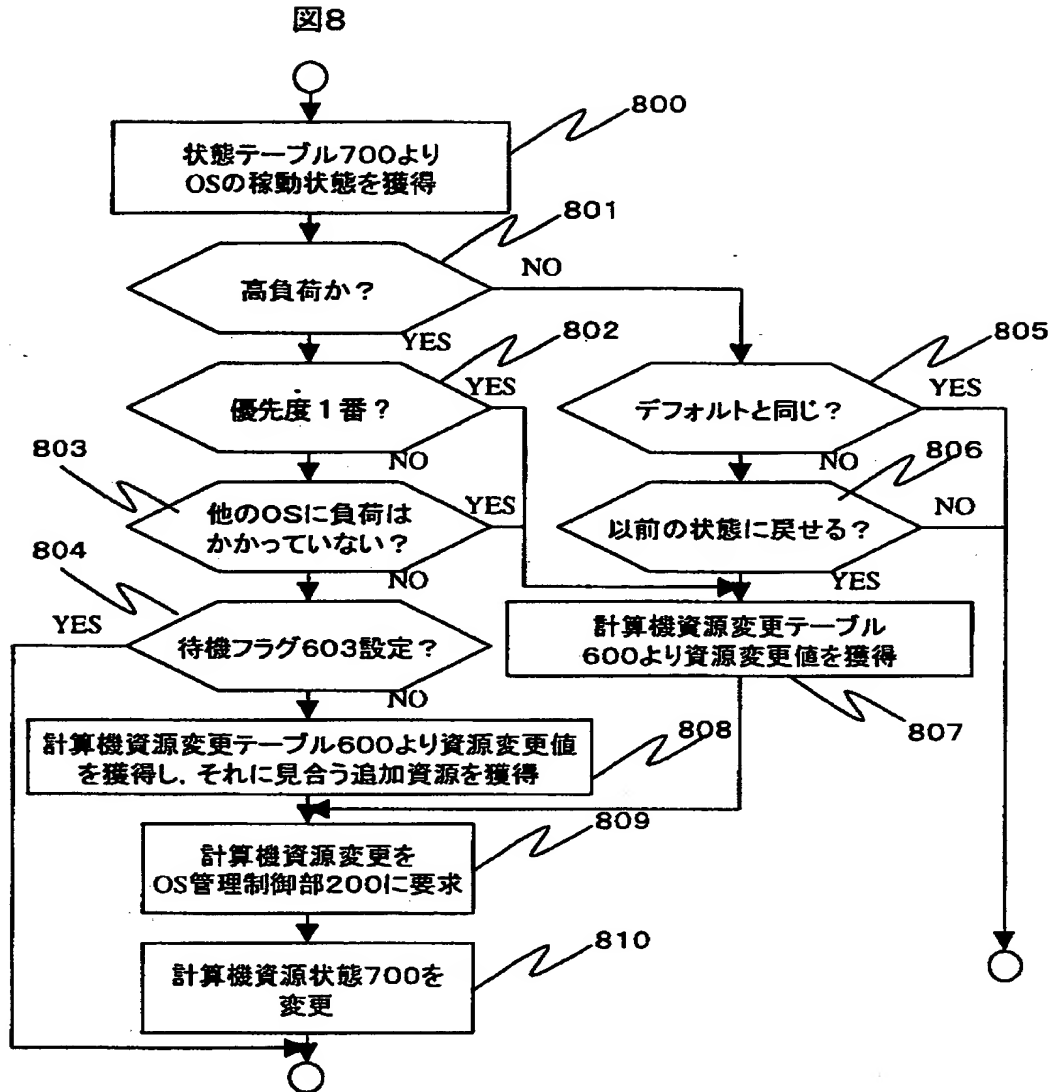
【図 7】

図 7

OS 名	CPU	CPU 負荷	主記憶	主記憶 負荷	I/O	I/O 負荷	I/O 2	I/O 2 負荷
OS 1								
OS 2								
OS n								

700: 計算機資源変更状態テーブル

【図8】



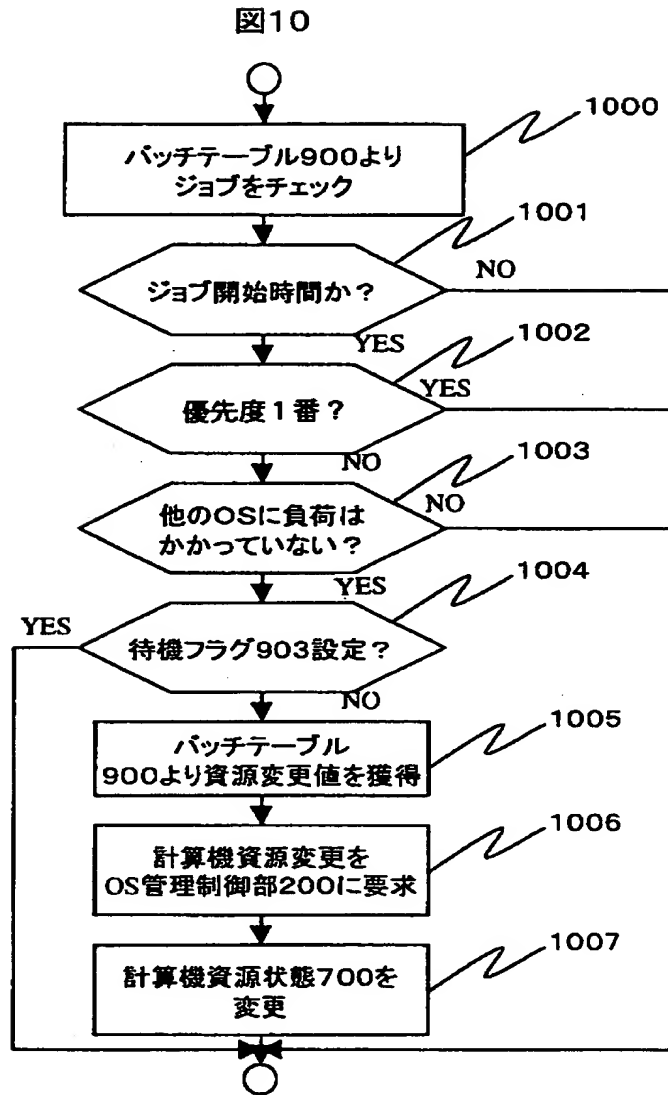
【図 9】

図9

900							
901							
902							
903							
904							
905							
906							
ジョブ 名	OS名	ジョブ 開始時 間	ジョブ 終了時 間	待 機 フ ラ グ	計 算 機 資 源 要 求 値		
					907 CPU	908 主記憶	909 I/O
1							
2							
n							

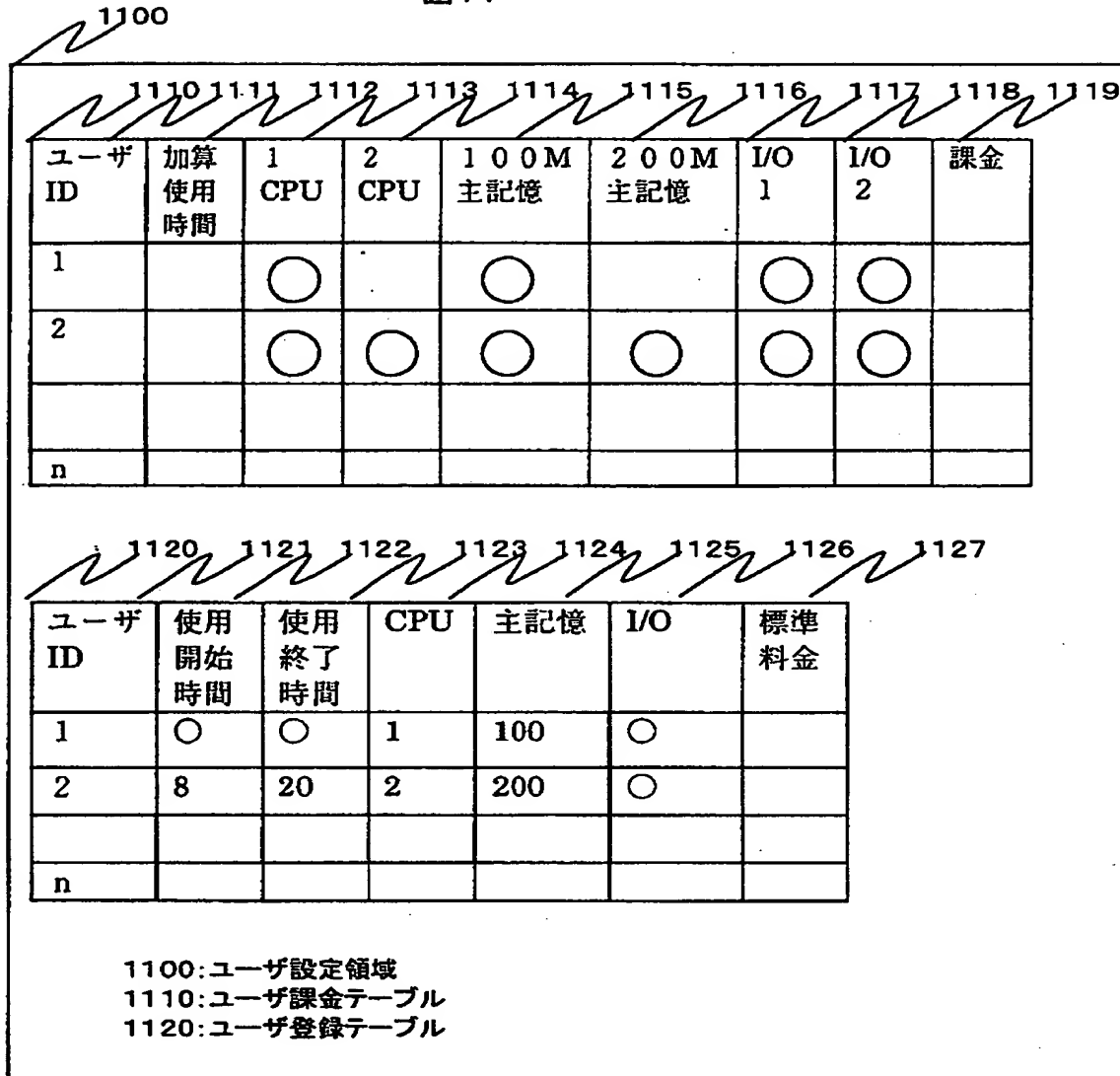
900:計算機資源変更テーブル2

【図 1 0】



【図 1 1】

図 11



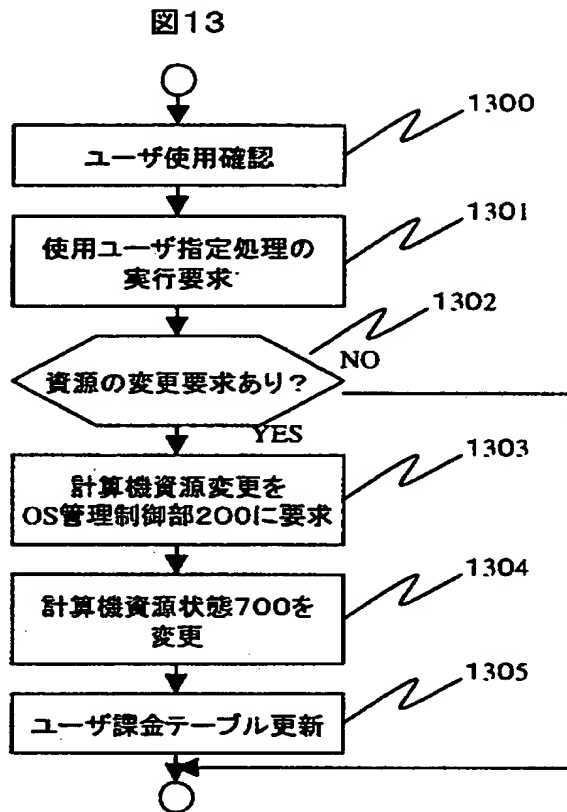
【図 1 2】

図12

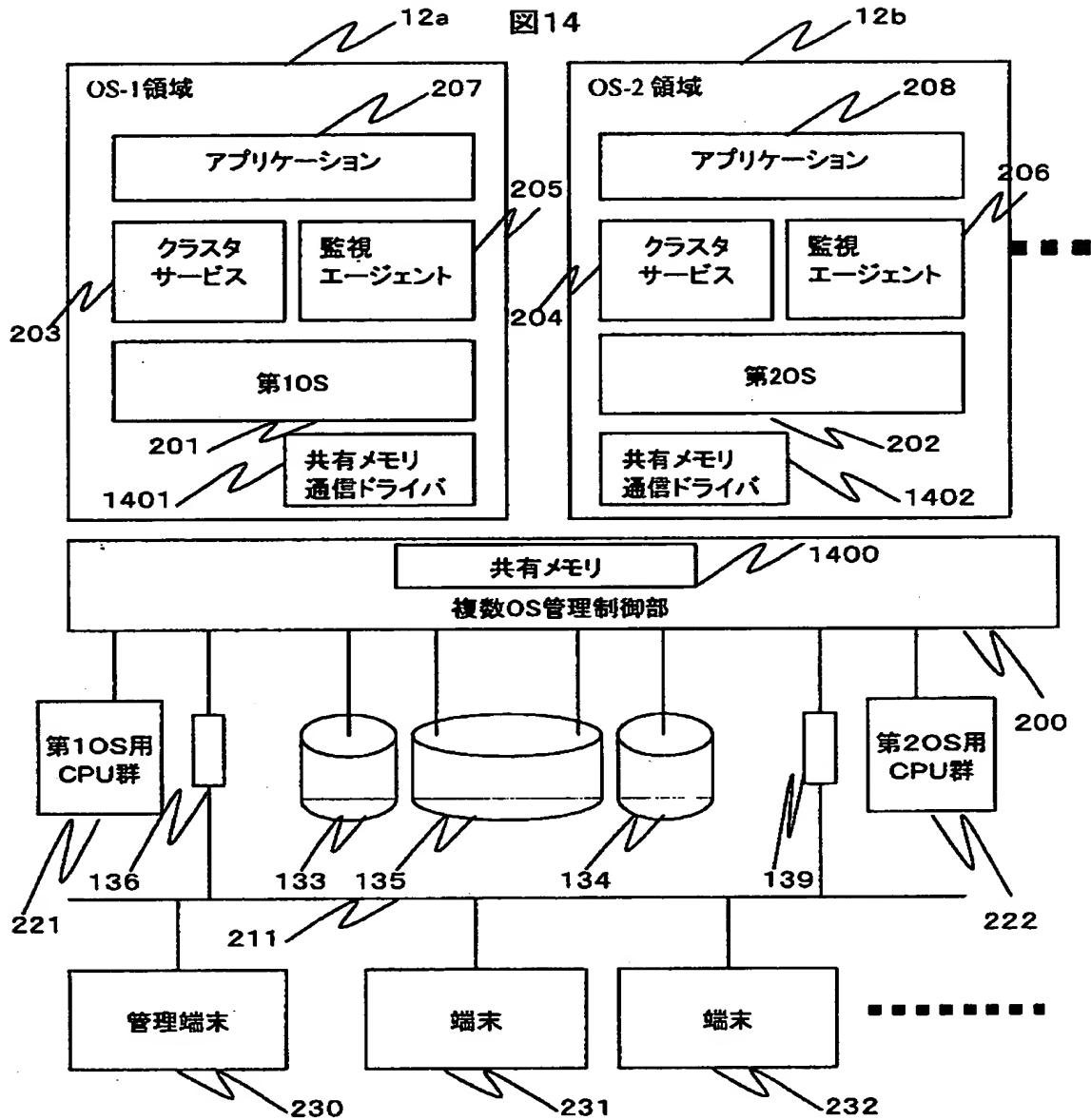
1200				
1210	1220	1221	1222	122n
使用時間	1 CPU	2 CPU		N CPU
1				
2				
1230 1240 1241 1242 124n				
使用時間	1 0 0 MB	2 0 0 MB		N MB
1				
2				
1250 1260 1261 1262 126n				
使用時間	1 G	2 G		N G
1				
2				

1200: 計算機資源課金基準領域
 1210: CPU課金基準テーブル
 1230: 主記憶課金基準テーブル
 1250: 外部装置課金基準テーブル

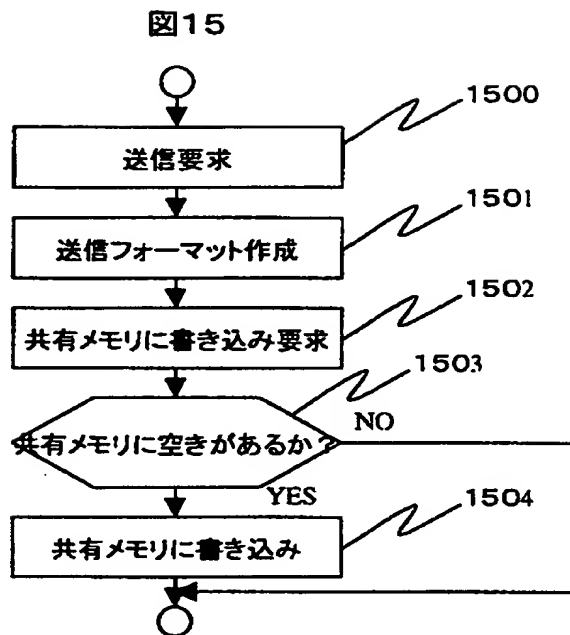
【図 1 3】



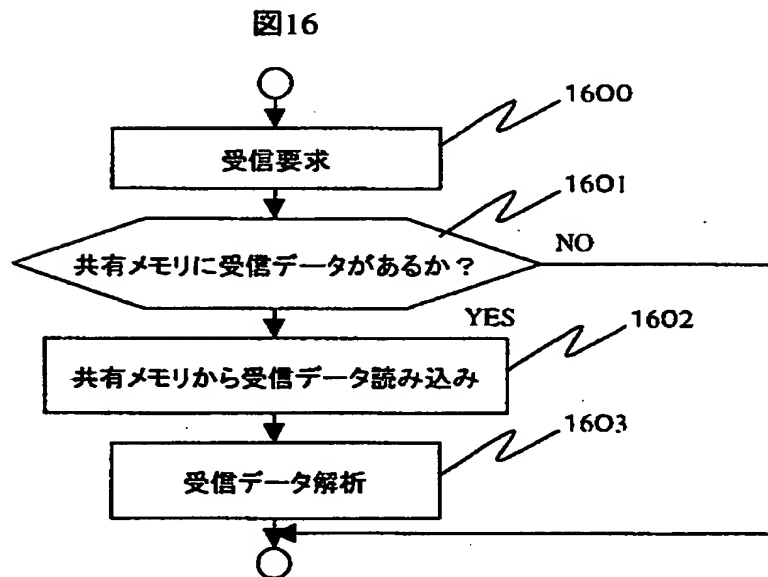
【図14】



【図 15】



【図 16】



【書類名】 要約書

【要約】

【課題】 複数のOSが用いる計算機資源を、各OSの状況に応じて効率的に割り当て制御する。

【解決手段】 1 台の計算機システム内に複数のOSを稼働させ、各OSの計算機資源の変更、再配置を行うことが可能にクラスタシステムが構成されている。各OSを運用系、待機系として運用している場合、OS管理制御部200は、各OSの状況を監視して、運用系OSの障害時に計算機資源を別の正常なOSに多く割り当てて、それを運用系とする。これにより、障害に関わらず、計算機システムの処理能力を変えことなく計算機システムを稼働させることができる。また、OS管理制御部200は、各OSの負荷状態を監視して、その負荷に応じて、計算機資源を割り当てることが可能である。これにより、OS間の処理能力を必要に応じて調整することができる。

【選択図】 図2

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所